WP. 27
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)

Joint UNECE/Eurostat work session on statistical data confidentiality
(Geneva, Switzerland, 9-11 November 2005)

Topic (iv): Access to business microdata for analysis

METHODS OF SECURE COMPUTATION AND DATA INTEGRATION

Invited Paper

Submitted by the National Institute of Statistical Sciences, University of Cincinnati, Duke University and Bristol-Myers Squibb, United States of America[1]

[1] Prepared by Alan F. Karr, Xiaodong S. Lin, Jerome P. Reiter and Ashish P. Sanil.

Methods of Secure Computation and Data Integration

Alan F. Karr*, Xiaodong S. Lin**, Jerome P. Reiter***, Ashish P. Sanil****

* National Institute of Statistical Sciences, Durham, NC, USA (karr@niss.org).

** University of Cincinnati, Cincinnati, OH, USA (xiaodong.lin@uc.edu).

*** Duke University, Durham, NC, USA (jerry@stat.duke.edu).

**** Bristol-Myers Squibb, Princeton, NJ, USA.

Abstract. Reluctance of statistical agencies and other data owners to share their possibly confidential or proprietary data with others who own related databases is a serious impediment to conducting mutually beneficial analyses. This paper reviews methods for secure computation that potentially allow agencies to share data without compromising data confidentiality. The methods discussed include secure summation protocols, secure matrix product protocols, and synthetic data approaches.

1 Introduction

In many contexts, statistical agencies, survey organizations, businesses, and other data owners (henceforth all called agencies, to save writing) with related databases can benefit by integrating their data. For example, statistical models can be fit using more records or more attributes when databases are combined than when databases are analyzed separately. However, agencies may not be able or willing to combine their databases because of concerns about data confidentiality. These concerns can be present even when the agencies cooperate: all may wish to perform integrated analyses, but no one wants to break the confidentiality of others' data. In this paper, we review some approaches to data integration that aim to limit the risks of disclosures while maintaining the utility of the integrated data. In particular, we review secure computation techniques and approaches based on synthetic, i.e. simulated, data.

Data integration can be categorized into two general settings. Horizontally partitioned databases comprise the same attributes for disjoint sets of data subjects. For example, several local educational agencies might want to combine their students' data to improve the precision of analyses of the general student population. Vertically partitioned databases comprise the same data subjects, but each database contains different sets of attributes. For example, one agency might have employment information, another health data, and a third information about education,

all for the same individuals. A statistical analysis predicting health status from all three sources of attributes is more informative than, or at least complementary to, separate analyses from each data source.

Various assumptions are possible about the participating agencies, for example, whether they use "correct" values in the computations, follow computational protocols, or collude against one another. We assume the agencies wish both to cooperate and to preserve the privacy of their individual databases. We assume that the agencies are "semi-honest:" each follows the agreed-on computational protocols properly, but may retain the results of intermediate computations. The results of analyses of horizontally or vertically partitioned data are to be shared among all participating agencies and possibly disseminated to the broader public.

2 Horizontally partitioned data

Several algorithms have been developed for performing secure analyses of horizontally partitioned data. Among them, Evfimievski *et al.* (2004) and Kantarcioglu and Clifton (2002) present methods for data mining with association rules; Lin *et al.* (2005) present methods for model based clustering; and, Karr *et al.* (2005b,c) present methods for secure regression analyses, including model diagnostics. The literature on privacy-preserving data mining (Lindell and Pinkas, 2000; Agrawal and Srikant, 2000) contains related results. Here we summarize the approach of Karr *et al.* (2005b,c), who use the secure summation protocol (Benaloh 1987) to perform regression and other analyses on horizontally partitioned databases.

2.1 Secure summation protocol

Consider $K > 2$ cooperating, semi-honest agencies, such that Agency j has a value v_j. The agencies wish to compute $v = \sum_{j=1}^{K} v_j$ so that each Agency j learns only the minimum possible about the other agencies' values, namely the value of $v_{(-j)} = \sum_{\ell \neq j} v_\ell$. The secure summation protocol (Benaloh 1987) can be used to effect this computation.

Following the presentation in Karr *et al.* (2005b), let m be a very large number—which is known to all the agencies—such that $0 \leq v < m$. One agency is designated the master agency and numbered 1. The remaining agencies are numbered $2, \ldots, K$. Agency 1 generates a random number R from $[0, m)$. Agency 1 adds R to its local value v_1 and sends the sum $s_1 = (R + v_1) \bmod m$ to Agency 2. Since the value R is chosen randomly from $[0, m)$, Agency 2 learns nothing about the actual value of v_1.

For the remaining agencies $j = 2, \ldots, K-1$, the algorithm is as follows. Agency j receives

$$s_{j-1} = (R + \sum_{s=1}^{j-1} v_s) \bmod m,$$

from which it can learn nothing about the actual values of $v_1, \ldots, v_{j-1}$. Agency j

then computes and passes on to Agency $j+1$

$$s_j = (s_{j-1} + v_j) \bmod m = (R + \sum_{s=1}^{j} v_s) \bmod m.$$

Finally, agency K adds v_K to $s_{K-1} (\bmod m)$, and sends the result s_K to agency 1. Agency 1, which knows R, then calculates v by subtraction:

$$v = (s_K - R) \bmod m$$

and shares this value with the other agencies.

For cooperating, semi-honest agencies, the use of arithmetic mod m may be superfluous. It does, however, provide one layer of additional protection: without it, a large value of s_2 would be informative to Agency 2 about the value of R.

This method for secure summation faces an obvious problem if some agencies collude. For example, agencies $j-1$ and $j+1$ can together compare the values they send and receive to determine the exact value for v_j. Secure summation can be extended to work for an honest majority. Each agency divides v_j into shares. The sum for each share is computed individually. However, the path used is altered for each share so that no agency has the same neighbor twice. To compute v_j, the neighbors of agency j from every iteration would have to collude.

2.2 Secure regression via secure summation

Suppose the agencies wish to combine their data to fit the usual linear regression model:

$$Y = X\beta + \epsilon, \tag{1}$$

where

$$X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np-1} \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \tag{2}$$

and

$$\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}. \tag{3}$$

Under the condition that $\mathrm{Cov}(\varepsilon) = \sigma^2 I$, the least squares estimate for β is of course $\hat{\beta} = (X^T X)^{-1} X^T Y$.

When the data are horizontally partitioned across K agencies, each agency j has its own share of data

$$X^j = \begin{bmatrix} x^j_{11} & \dots & x^j_{1p} \\ \vdots & \ddots & \vdots \\ x^j_{n_j 1} & \dots & x^j_{n_j p} \end{bmatrix}, \qquad y^j = \begin{bmatrix} y^j_1 \\ \vdots \\ y^j_{n_j} \end{bmatrix}. \tag{4}$$

Here n_j denotes the number of data records for agency j.

Using (4) and altering indices as appropriate, we can rewrite (2) in partitioned form as

$$X = \begin{bmatrix} X^1 \\ \vdots \\ X^K \end{bmatrix} \qquad Y = \begin{bmatrix} Y^1 \\ \vdots \\ Y^K \end{bmatrix} \tag{5}$$

and (3) as

$$\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{p-1} \end{bmatrix} \qquad \epsilon = \begin{bmatrix} \varepsilon^1 \\ \vdots \\ \varepsilon^K \end{bmatrix}. \tag{6}$$

Note that β does not change.

To compute $\hat{\beta}$, it is necessary to compute $X^T X$ and $X^T Y$. Because of the partitioning in (5), this can be done locally and the results combined entry-wise using secure summation. Specifically,

$$X^T X = \sum_{j=1}^{K} (X^j)^T X^j. \tag{7}$$

Each agency j can compute locally its own $(X^j)^T X^j$, and the results can be added entry-wise using secure summation to yield $X^T X$, which then can be shared among all the agencies. Similarly, since

$$X^T Y = \sum_{j=1}^{K} (X^j)^T Y^j,$$

$X^T Y$ can be computed by local computation of the $(X^j)^T Y^j$ and secure summation. This provides all the pieces necessary for each agency to compute $\hat{\beta}$.

The least squares estimate of σ^2 also can be computed securely. Since

$$S^2 = \frac{(Y - X\hat{\beta})^T (Y - X\hat{\beta})}{n - p}, \tag{8}$$

and $X^T X$ and $\hat{\beta}$ have been computed securely, the only thing left is to compute n and $Y^T Y$, again using secure summation. The agencies then can compute the estimated covariance matrix of the $\hat{\beta}$, which equals $S^2 (X^T X)^{-1}$.

It is also possible to share via secure summation statistics useful for model diagnostics, including correlations between predictors and the residuals, the coefficient of determination R^2, and the hat matrix $X(X^T X)^{-1} X^T$. Values of residuals are risky to share, since they reveal information about the dependent variable. Karr *et al.* (2005b) describe an approach for simulating plots of residuals versus predictors that mimic the real-data plots, based on the techniques of Reiter (2003a), which can be used for model diagnostics without releasing genuine residuals.

3 Vertically partitioned data

For vertically partitioned data, secure analysis methods exist for association rule mining (Vaidya and Clifton, 2002), K-means clustering (Vaidya and Clifton, 2003), and linear discriminant analysis (Du *et al.*, 2004). Du *et al.* (2004) and Sanil *et al.* (2004) present approaches to computing regression coefficients in vertically partitioned data, using methods that do not share the sample mean and covariance matrix. Here we review the approach of Karr *et al.* (2005a), which assumes the agencies are willing to share sample means and covariances of the integrated database but not the raw data. For simplicity, we describe the secure computation protocol for matrix products as a two-agency protocol. It is readily applicable to multi-agency cases.

3.1 Secure matrix product protocol

Following Karr *et al.* (2005a), let Agency A possess p data vectors $\{X_1, X_2, \ldots, X_p : X_i \in \Re^n\}$ and Agency B have q vectors $\{Y_1, Y_2, \ldots, Y_q : Y_i \in \Re^n\}$. Let $X = [X_1, X_2, \ldots, X_p]$ and $Y = [Y_1, Y_2, \ldots, Y_q]$ denote the respective data matrices, and assume $p < q$. We assume the matrices are of full rank; if not, the agencies remove any linearly dependent columns. We also assume the attributes in X and Y are disjoint; if not, the agencies coordinate so that any common attributes are included in only one matrix. Lastly, we assume that X and Y have disjoint attributes (columns) but the same data subjects (rows).

Agency A and Agency B wish to compute securely the $(p \times q)$ matrix $X^T Y$ and share it. It is necessary for the participating agencies to align their common data subjects in the same order. We assume each agency possesses a primary key, for example social security numbers, that is shared to facilitate this ordering.

In the interest of fairness to each participating agency, and to encourage trust among the agencies, we desire a protocol for secure matrix products that is symmetric in the amount of information exchanged. That is, the agencies should learn roughly the same amount about each other's data from the information shared in the protocol. A protocol that accomplishes this approximately is described by following procedure:

1. Agency A generates a set of $g = \lfloor \frac{n-p}{2} \rfloor$ orthonormal vectors $\{Z_1, Z_2, \ldots, Z_g : Z_i \in \Re^n\}$ such that $Z_i^T X_j = 0$ for any i, j. Agency A then sends the matrix $Z = [Z_1, Z_2, \ldots, Z_g]$ to Agency B.

2. Agency B computes $W = (I - ZZ^T)Y$, where I is an identity matrix. Agency B sends W to Agency A.

3. Agency A calculates $X^T W = X^T(I - ZZ^T)Y = X^T Y$ since $X_j^T Z_i = 0$ for any i, j.

The vector dot-product protocol is a special case of the matrix product. A method for generating Z is presented in Karr *et al.* (2005a).

It might appear that Agency B's data can be learned exactly since Agency A knows both W and Z. However, W has rank $(n-g) = (n-2p)/2$, so that Agency A cannot invert it to obtain Y.

Exact data values are not revealed in this protocol, but each agency can learn about the others' data from the constraints on the data values imposed by the values of the shared statistics. For any matrix product protocol where X^TY is learned by all agencies, each agency knows at minimum pq constraints, i.e those implied by the values of X^TY. In addition, Agency A knows the g dimensional subspace that the Y_i lie in (as given by $W = (I - ZZ^T)Y$). Thus, Agency A has a total of $g + pq$ constraints on Y. Agency B knows the $(n-g)$ dimensional subspace that the X_i lie in (the subspace orthogonal to Z). Thus, Agency B has a total of $n - g + pq$ constraints on X.

In most settings involving vertically partitioned data, the $n \gg pq$, so that $g \approx \frac{n}{2}$. Hence, we can say that both agencies can place the other agencies' data in an approximately $\frac{n}{2}$ subspace, so that the protocol is approximately symmetric in the information shared.

The protocol is not immune to breaches of confidentiality if the agencies do not cooperate in a semi-honest fashion. For example, suppose Agency A sends to Agency B a Z such that $(I - ZZ^T)$ contains one column with all zeros except for a non-zero constant in one row. Agency A then learns the value of Agency B's data for the data subject in that row through X^TW. Other bogus Z could yield similar disclosures.

Even when the agencies are semi-honest, disclosures might be generated because of the values of the attributes themselves. As a simple example, suppose X includes a variable that equals zero for all but one of the data subjects. Even with a legitimate Z, the X^TY will reveal that subject's value of Y. Similar problems could arise when some X_i contains non-zeros for only a small number of records, particularly when reliable prior information on those records' values of some Y_j is known. For example, suppose two firms are the only ones in a certain industry in a certain city, with one being large and the other being small. Let X_i be an indicator with ones for those two firms and zeros for other firms. Let Y_j be some sensitive attribute positively correlated to the size of a firm. The $X_i^TY_j$ equals the sum of the two firms' values, but most of that sum is contributed by the large firm. Thus, $X_i^TY_j$ may be sufficiently close to the one firm's value of Y_j as to be a disclosure.

Disclosures resulting from subject matter considerations can be difficult to prevent. If Agency B does not know that Agency A has a variable like the X_i above, there is almost no way for Agency B to prevent disclosing some values in the matrix multiplications. A related problem occurs if one agency has attributes that are nearly linear combinations of the other agency's attributes. When this happens, accurate predictions of the data subjects' values can be obtained from linear

regressions built from the securely computed matrix products.

3.2 Linear Regression with arbitrary subsets of attributes

In this section, we apply the secure matrix product protocol to conduct secure linear regression analyses. Let the matrix of all variables in the possession of the agencies be $D = [D_1, \cdots, D_p]$, with

$$D_i = \begin{pmatrix} d_{i1} \\ \vdots \\ d_{in} \end{pmatrix}, \quad 1 \leq i \leq p \ . \tag{9}$$

The data matrix D is distributed through K agencies: $A_1, A_2, \cdots, A_K$. Each agency, A_j, possesses p_j disjoint columns of D, where $\sum_K p_j = p$.

A regression model of some dependent variable, say $D_i \subset D$, on a collection of the other variables, say $D_0 \subseteq D - D_i$, is of the form

$$D_i = D_0\beta_0 + \epsilon_0 \tag{10}$$

where $\epsilon_0 \sim N(0, \sigma_0^2)$. Typically, the model includes an intercept term. This is achieved by including a column of ones in D_0. Without loss of generality, we assume that $D_1^T = (1, 1, \ldots, 1)$ and that it is owned by Agency A_1.

Our goal is to regress any D_i on some arbitrary subset D_0 using secure computations. It is well known that the maximum likelihood estimates of σ_0^2 and β_0, as well as the standard errors of the estimated coefficients, can be easily obtained from the sample covariance matrix of D, for example using the sweep algorithm (Beaton, 1964). Hence, the agencies need only the elements of the sample covariance matrix of D to perform the regression. Each agency computes and shares the block-diagonal elements of the matrix corresponding to its variables, and the agencies use secure matrix computations to compute the off-diagonal elements, thus completing the sample covariance matrix.

The types of model diagnostic measures available in vertically partitioned data settings depend on the amount of information the agencies are willing to share. Diagnostics based on residuals require the predicted values, $D_0\hat{\beta}_0$. These can be obtained using the secure matrix product protocol, since

$$D_0\hat{\beta}_0 = D_0(D_0{}^T D_0)^{-1} D_0{}^T D_i. \tag{11}$$

Alternatively, once the $\hat{\beta}_0$ is shared, each agency could compute the portion of $D_0\hat{\beta}_0$ based on the variables in its possession, and the vectors can be summed across agencies using the secure summation protocol outlined in Section 2.1.

Once the predicted values are known, the agency with the dependent variable D_i can calculate the residuals $E_0 = D_i - D_0\hat{\beta}_0$. If that agency is willing to share the

residuals with the other agencies, each agency can perform plots of residuals versus its independent variables and report the nature of any lack of fit to the other agencies. Sharing E_0 also enables all agencies to obtain Cook's distance measures, since these are solely a function of E_0 and the diagonal elements of $H = D_0({D_0}^T D_0)^{-1}{D_0}^T$, which can be securely computed.

The agency with D_i may be unwilling to share E_0 with the other agencies, since sharing essentially reveals the values of D_i. In this case, one option is to compute the correlations of the residuals with the independent variables using the secure matrix product protocol. Additionally, the agency with D_i can make a plot of E_0 versus $D_0\hat{\beta}_0$, and a normal quantile plot of E_0, and report any evidence of model violations to the other agencies. The number of residuals exceeding certain thresholds, i.e. outliers, also can be reported.

3.3 Synthetic data approach for vertically partitioned data

The secure matrix protocol requires that agencies pre-specify the regression analyses of interest. In some settings this could be problematic. For example, it may be necessary to transform some variables to obtain a regression that fits the data appropriately. Agencies can apply the secure matrix protocol more than once, e.g. on the original data to enable model checking and then on transformed data to improve the model, but repeating the protocol generates additional constraints on X and Y that reduce confidentiality protection.

To introduce flexibility of modeling, Kohnen and Reiter (2004) and Kohnen (2005) propose that agencies share synthetic, i.e. simulated, data that mimic the relationships in the real data. To motivate their idea, consider the case where Agency A is willing to share its X with Agency B, but Agency B is not willing to share its Y with Agency A. The approach proceeds as follows:

1. Agency A sends X to Agency B.

2. Agency B fits a model $f(Y|X)$ that relates Y to X, based on the passed X and its genuine Y.

3. Agency B simulates a new value of Y from the model $f(Y|X)$ and passes these simulated data to Agency A. Agency B repeats this M times, so that M versions of the synthetic Y are passed to Agency A.

4. Agency A analyzes the M datasets formed by combining the X with each version of Y using the methods for analyzing multiply-imputed, partially synthetic datasets (Reiter, 2003b).

At stage 2, Agency B can either (i) send $f(Y|X)$ and its parameters to Agency A, or (ii) simulate new values of Y from the model $f(Y|X)$ and pass these simulated values to Agency A. The latter strategy is preferred when the model and its parameters

represent a disclosure risk (e.g., parameters in log-linear models for categorical data correspond to cell counts in tables, which may be sensitive) or when the model is too complicated to send (e.g., a semi-parametric model). We assume that Agency B will generate and pass new values of Y.

The multiple versions of Y are needed to enable Agency A to estimate uncertainties in parameter estimates correctly. One version of Y is insufficient, because the process of drawing values of Y from a distribution introduces additional variability into parameter estimates that is not easily estimated from one dataset. The prescription for releasing multiple copies follows the rationale for generating multiply-imputed, partially synthetic datasets (Reiter, 2003b, 2004).

As an extension to this case, the agencies may be willing to share X with each other but not with the broader public. To release data to the public, Agency A can simulate completely synthetic data (Raghunathan, Reiter, Rubin, 2003). That is, it can simulate values of X and values of Y using its original values of X and the simulated values of Y it received from Agency B. Methods for doing this, as well as methods for obtaining inferences from such datasets, are described by Kohnen (2005).

We next move to the more general case where Agency A is not be willing to share X with Agency B. The key difference in the algorithm is in step 2, since X cannot be passed without some way of protecting it. Kohnen (2005) proposes that Agency A generate disguiser copies of X–that is, new values of X that mimic the distribution of the genuine X–and send them to Agency B along with the genuine X. Agency B then fits models for $Y|X$ for each of the copies of X and sends simulated values of Y back for each back to Agency A. Agency A discards all the simulated Y except for the ones that correspond to the genuine X. With L perfect disguisers, Agency B has a $1/L$ chance of guessing which of the L datasets contains the true X. For sufficiently large L, this may provide adequate protection.

Obviously, the protection of X is compromised if Agency B can distinguish the genuine X from the disguisers. This could be accomplished if Agency B knows certain values of X and therefore can hunt for them in the passed copies. To prevent this in a semi-honest setting, Agency B can tell Agency A which values it has, so that these values can be included in all passed copies of X. Agency B also might be able to determine the genuine X by looking for unusual results in the various versions of $f(Y|X)$. For example, it may be the case that the genuine X has the strongest correlations with Y. Kohnen (2005) describes several such risks, as well as some methods for reducing them.

Ideally, the disguiser X values are generated from $f(X|Y)$. This is not easy to do, since Agency A does not know Y. It may be possible to approximate this distribution, perhaps using methods from standard disclosure limitation strategies. Research on generating good disguisers is a high priority item for this approach.

4 Conclusion

In this paper, we summarized several approaches to secure data integration. These approaches generate many practical challenges, which as of this writing have not been fully met. Some of these include:

- How do we specify models without viewing the data, which is implicit in the secure computation methods?
- How do we perform secure computation for models that don't have sums and products as sufficient statistics?
- How do we incorporate errors when matching records in vertically partitioned data?
- How do we account for differences in data quality and definitions?
- How do we account for disclosure risks from models that fit too well?

Statisticians have only recently started investigating the data integration setting (computer scientists have been active in this area for longer). It is an area that is likely to grow in relevance, as data owners of all types seek to gain the benefits from data integration. And, as the questions listed above indicate, it is an area rich in topics for statistical research.

Acknowledgements

This research was supported by NSF grant IIS–0131884 to the National Institute of Statistical Sciences.

References

Agrawal, R. and Srikant, R. (2000) "Privacy-Preserving Data Mining", in *Proceedings of the 2000 ACM SIGMOD on Management of Data*, 439–450.

Beaton, A. (1964) "The Use of Special Matrix Operations in Statistical Calculus", *Research Bulletin RB-64-51*, Educational Testing Service, Princeton, NJ.

Benaloh, J. (1987) "Secure Sharing Homomorphisms: Keeping Shares of a Secret Secret", in *Advances in Cryptography: CRYPTO86*, ed. A. M. Odlyzko, New York: Springer-Verlag, **263**, 251–260.

Du, W., Han, Y., and Chen, S. (2004) "Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification", in *Proceedings of the 4th SIAM Conference on Data Mining*, 222–233.

Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. (2004) "Privacy-Preserving Mining of Association Rules", *Information Systems*, **29**, June 2004.

Kantarcioglu, M. and Clifton, C. (2002) "Privacy-Preserving Distributed Mining of Association Rules on Horizontally-Partitioned Data", in *Proceedings of Eigth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Karr, A., Lin, X., Sanil, A., and Reiter, J. (2005a) "Privacy-Preserving Analyses of Vertically Partitioned Data Using Secure Matrix Product Protocols", Technical Report, National Institute of Statistical Sciences.

Karr, A., Lin, X., Sanil, A., and Reiter, J. (2005b) "Secure Regressions on Distributed Databases", *Journal of Computational and Graphical Statistics*, **14**, 263–279.

Karr, A., Lin, X., Sanil, A., and Reiter, J. (2005c) "Secure Statistical Analyses of Distributed Databases." in *Statistical Methods in Counterterrorism*, ed. D. Olwell and A. Wilson, ASA-SIAM Series on Statistics and Applied Probability, to appear.

Karr, A., Feng, J., Lin, X., Sanil, A., Young, S., and Reiter, J. (2005) "Secure Analyses of Distributed Chemical Databases without Data Integration", *Journal of Computer-Aided Molecular Design*, to appear.

Kohnen, C. (2005) "Using Multiply-Imputed, Synthetic Data to Facilitate Data Sharing", PhD Dissertation, Institute of Statistics and Decision Sciences, Duke University.

Kohnen, C. and Reiter, J. (2004) "Sharing Confidential Data Among Multiple Agencies Using Multiply-Imputed, Synthetic Data", *Proceedings of the Joint Statistical Meetings*, American Statistical Association.

Lin, X., Clifton, C., and Zhu, Y. (2004) "Privacy-Preserving Clustering with Distributed EM Mixture Models. *Knowledge and Information Systems*, **8**, 68–81.

Lindell, Y. and Pinkas, B. (2000) "Privacy-Preserving Data Mining" in *Advances in Cryptology: Crypto2000*, New York: Springer-Verlag, **1880**, 36–54.

Raghunathan, T., Reiter, J., and Rubin, D. (2003) "Multiple Imputation for Statistical Disclosure Limitation" *Journal of Official Statistics*, **19**, 1–16.

Reiter, J. (2003a) "Model Diagnostics for Remote Access Regression Servers", *Statistics and Computing*, **13**, 371–380.

Reiter, J. (2003b) "Inference for Partially Synthetic, Public Use Microdata Sets", *Survey Methodology*, **29**, 181–188.

Reiter, J. (2004) "Simultaneous Use of Multiple Imputation for Missing Data and Disclosure Limitation", *Survey Methodology*, **30**, 235–242.

Sanil, A., Karr, A., Lin, X., and Reiter, J. (2004) "Privacy-Preserving Regression Modeling Via Distributed Computation." *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 677–682.

Vaidya, J. and Clifton, C. (2002) "Privacy-Preserving Association Rule Mining Over Vertically Partitioned Data", *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 639–644.

Vaidya, J. and Clifton, C. (2003) "Privacy-Preserving k-Means Clustering Over Vertically Partitioned Data", *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.